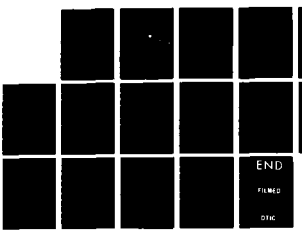
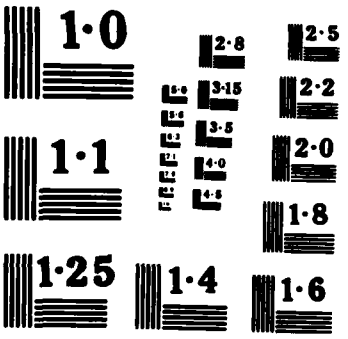
AD-A158 541 STOP AND LOOK DETECTION ALGORITHM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA A F ANDRUS MAY 85 NPS55-85-011 1/1

UNCLASSIFIED F/G 9/2 NL





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

(2)

NPS55-85-011

NAVAL POSTGRADUATE SCHOOL

Monterey, California

STOP AND LOOK DETECTION ALGORITHM

by

Alvin F. Andrus

May 1985

Prepared for:
Naval Postgraduate School
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-5100

Rear Admiral R. H. Shumaker
Superintendent

David A. Schrady
Provost

Reproduction of all or part of this report is authorized.

This report was prepared by: Alvin F. Andrus

Alvin F. Andrus
ALVIN F. ANDRUS
Associate Professor of Operations Research

Reviewed by:

Alan R. Washburn
ALAN R. WASHBURN, Chairman
Department of Operations Research

Released by:

Kneale T. Marshall
KNEALE T. MARSHALL
Dean of Information & Policy

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS55-85-011	2. GOVT ACCESSION NO. AD-A158541	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STOP AND LOOK DETECTION ALGORITHM		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Alvin F. Andrus		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5100		12. REPORT DATE May 1985 13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Search
Detection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Stop and Look Detection Algorithm is a procedure for computing the cumulative probability of detection as a function of time for a searcher looking discretely for an evading target. The assumptions required for computation are: target detection is deterministic, i.e., cookie cutter, the target leaves datum on a random fixed course at constant velocity, the searcher travels at constant velocity and the searcher stops and looks for the target at predetermined search points. This report contains a description of the algorithm, the IBM-PC BASIC Program for computation and several

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

examples. Additional keywords: BASIC programming language; computer graphics

TABLE OF CONTENTS

1. Problem Description:

The general search problem of interest in this report is one in which an evading target leaves datum on a random fixed course with constant velocity and the searcher begins searching for the target at a specified time late with constant velocity and at predetermined search points. At each search point the searcher stops and looks for the target. At each search point a detection range is used for target detection. If the target is within detection range of the searcher at the search point then the target is detected and the search is ended. If the target is not within detection range of the searcher at the search point then the target is not detected and the search continues with the searcher moving to the next search point. If the target is not detected at the last search point then the search ends.

2. Assumptions:

The assumptions used for computation in the Stop and Look Detection Algorithm are:

a. The target leaves datum with a constant velocity at time = 0.

b. The target leaves datum on a random fixed course uniformly distributed over the range (0,2*Pi).

c. The searcher begins searching at a specified time late with a constant velocity.

d. The searcher stops and looks for the target at predetermined search points.

e. Detection is deterministic, i.e., cookie cutter.

3. Input Requirements:

The coordinate system for describing the search is a two dimensional (X,Y) system centered on the datum located at (0,0). Within this framework the inputs required for the Stop and Look Detection Algorithm computation are:

a. The search points,

b. The searcher velocity,

c. The detection range,

d. The time late,

e. The target velocity.

The units used for all inputs must be consistent.

4. Modeling Procedures:

The Stop and Look Detection Algorithm computes the probability of detecting the target by the searcher by evaluating the intersection of the target location circle and the searcher detection circle for each search point and totaling the nonoverlapping range of target course values for which a detection is possible. The ratio of this range to the total range of target course values is the probability of detection.

When the searcher is located at search point (X,Y) at time T the target location circle is centered at (0,0) with radius T*(target speed) and the searcher detection circle is centered at (X,Y) with the search detection range as it's radius.

5. Stop and Look Detection Algorithm Description

The Stop and Look Detection Algorithm is described as follows. For each search point compute:

a. A(i) = (A1(i) , A2(i)).

A(i) is the set of target course values for which detection will occur at search point i. A(i) is determined by the intersection of the target location circle and the searcher detection circle.

b. B(i) = UNION A(j) for j = 1 to i.

B(i) = [(B1i(1),B2i(1)),(B1i(2),B2i(2)),...,(B1i(k),B2i(k))]

where

B1i(j) < B2i(j) < B1i(j+1).

B(i) is the set of nonoverlapping target course values for which detection will occur at search point i.

c. C(i) = SUM (B2i(j)-B1i(j))/(2*Pi) for j=1 to k.

C(i) is the cumulative probability of detection obtained by the stop and look at search point i.

The cumulative probability of detection is then computed as C(n). The Stop and Look Detection Algorithm computes the total measure of target course values for which detection will occur for all of the search points. Using the assumption that the target leaves datum on a random fixed course with constant velocity, the probability of detection is computed as the ratio of this total target course measure to 2*Pi.

6. BASIC Program Description:

The IBM-PC BASIC program for computing the probability of detection using the Stop and Look Detection Algorithm is included as Appendix A. The program includes initial screen output information that identifies the author and provides a general description of the search problem. Input file and output file names are provided by the user and must conform to BASIC file name conventions. The program uses a polar coordinate system with the datum (0,0) as the center. Computation for the intersection of the target location circle and the searcher detection circle is made in polar coordinates.

Inputs for the program are provided by menu controlled responses which will either create a new input file, use an existing input file or change and use an existing input file.

Outputs from the program are provided on the screen and include the date, program execution time, the input values, and for each search point the radius of the target location circle, A(i), B(i) and C(i). If B(i)=B(i-1) then B(i) is not included in the output.

When there are search points for which the target location circle is contained entirely within the searcher detection circle then the program output identifies these search points by displaying A(i)=(0,2*Pi). In this case the probability of detection is 1.0.

The output is also provided in an ASCII formatted output file. This output file may be printed on any standard printer.

The Stop and Look Detection Algorithm program, sample input and output files reside as ASCII formatted files on disk on the Naval Postgraduate School mainframe IBM-3033 computer. For users with access to this computer these files may be transfered. Information required for this transfer is:

User Identification Number: 0105P

Read Only Password: Andrus

File Name	File Type
STOPLOOK	BASIC
STOPLOOK	INPUT
STOPLOOK	OUTPUT

7. Example 1: Expanding Spiral Search:

This example computes the probability of detection for the searcher using an expanded spiral search procedure. The searcher begins the search at datum at time late with successive search points located around the datum in an expanded spiral. The input values for this example are:

Searcher Speed: 200 miles per hour
Target Speed: 30 miles per hour
Time late: 2 hours
Detection Range: 55 miles
Number of search points: 21
Search points:

i	(X,Y)	i	(X,Y)	i	(X,Y)
1	(0, 0)	8	(0, 100)	15	(0, 200)
2	(100, 0)	9	(100,-100)	16	(-100, 200)
3	(100, 100)	10	(200,-100)	17	(-200, 200)
4	(0, 100)	11	(200, 0)	18	(-200, 100)
5	(-100, 100)	12	(200, 100)	19	(-200, 0)
6	(-100, 0)	13	(200, 200)	20	(-200,-100)
7	(-100,-100)	14	(100, 200)	21	(-200,-200)

Table 1 contains the Stop and Look Detection Algorithm BASIC program output for this example.

Figure 1.1 contains an area coverage plot of this example data including the associated search point searcher detection circle, the intersecting arc of the target location circle, and the gaps in coverage of the target course values.

Figure 1.2 contains a plot of the cumulative probability of detection as a function of search time for this example.

From Table 1 and Figure 1.1 it can be seen that the search effort at search points 1, 8, 15, 16, 18-21 is wasted and the effort at search points 10, 11, 13, 17 is duplicated.

8. Example 2: Expanding Spiral Search:

This example uses the same data as Example 1 with the searcher speed increased to 290 miles per hour.

Table 2 contains the Stop and Look Detection Algorithm BASIC program output for this example.

Figure 2.1 contains an area coverage plot of this example data including the associated search point searcher detection circle, the intersecting arc of the target location circle, and the gaps in coverage of the target course values.

Figure 2.2 contains a plot of the cumulative probability of detection as a function of search time for this example.

From Table 2 and Figure 2.1 it can be seen that the search effort at search points 1, 10, 13 is wasted and the effort at search points 11,12, 15-21 is duplicated.

9. Example 3: Grid Search:

This example computes the probability of detection for the searcher using a grid search procedure. The input values for this example are:

Searcher Speed:	300 miles per hour
Target Speed:	30 miles per hour
Time late:	2 hours
Detection Range:	55 miles
Number of search points:	20

Search points:

i	(X,Y)	i	(X,Y)	i	(X,Y)
1	(-100, 200)	8	(0, 0)	15	(100,-200)
2	(-100, 100)	9	(0, 100)	16	(200,-200)
3	(-100, 0)	10	(0, 200)	17	(200,-100)
4	(-100,-100)	11	(100, 200)	18	(200, 0)
5	(-100,-200)	12	(100, 100)	19	(200, 100)
6	(0,-200)	13	(100, 0)	20	(200, 200)
7	(0,-100)	14	(100,-100)		

Table 3 contains the Stop and Look Detection Algorithm BASIC program output for this example.

Figure 3.1 contains an area coverage plot of this example data including the associated search point searcher detection circle, the intersecting arc of the target location circle, and the gaps in coverage of the target course values.

Figure 3.2 contains a plot of the cumulative probability of detection as a function of search time for this example.

From Table 3 and Figure 3.1 it can be seen that the search effort at search points 1, 2, 5, 6, 8, 11, 13, 16 is wasted and the effort at search point 20 is duplicated.

Andrus Stop-Look Detection Algoritha 05-24-1985

NUMBER OF (X,Y) = 21
TIME LATE = 2.00
SEARCH VELOCITY = 200.00
TARGET VELOCITY = 30.00
DETECTION RANGE = 55.00

I : SEARCH POINT NUMBER
X,Y : SEARCH POINT COORDINATES
TIME : TIME SEARCHER AT (X,Y)
TLCR : TARGET LOCATION CIRCLE RADIUS
A1,A2: TARGET/SEARCH CIRCLE INTERSECTION ANGLES
C : CUMULATIVE PROBABILITY OF DETECTION

I	X	Y	TIME	TLCR	A1	A2	C
1	0.00	0.00	2.00	60.00	0.00	0.00	0.00
2	100.00	0.00	2.50	75.00	5.71	0.57	0.18
3	100.00	100.00	3.00	90.00	0.61	0.96	0.24
4	0.00	100.00	3.50	105.00	1.03	2.11	0.41
5	-100.00	100.00	4.00	120.00	1.96	2.75	0.51
6	-100.00	0.00	4.50	135.00	2.77	3.51	0.63
7	-100.00	-100.00	5.00	150.00	3.55	4.30	0.75
8	0.00	-100.00	5.50	165.00	0.00	0.00	0.75
9	100.00	-100.00	6.00	180.00	5.25	5.74	0.82
10	200.00	-100.00	6.50	195.00	5.59	6.04	0.82
11	200.00	0.00	7.00	210.00	6.02	0.26	0.82
12	200.00	100.00	7.50	225.00	0.22	0.71	0.83
13	200.00	200.00	8.00	240.00	0.65	0.92	0.83
14	100.00	200.00	8.50	255.00	0.92	1.30	0.84
15	0.00	200.00	9.00	270.00	0.00	0.00	0.84
16	-100.00	200.00	9.50	285.00	0.00	0.00	0.84
17	-200.00	200.00	10.00	300.00	2.18	2.54	0.84
18	-200.00	100.00	10.50	315.00	0.00	0.00	0.84
19	-200.00	0.00	11.00	330.00	0.00	0.00	0.84
20	-200.00	-100.00	11.50	345.00	0.00	0.00	0.84
21	-200.00	-200.00	12.00	360.00	0.00	0.00	0.84

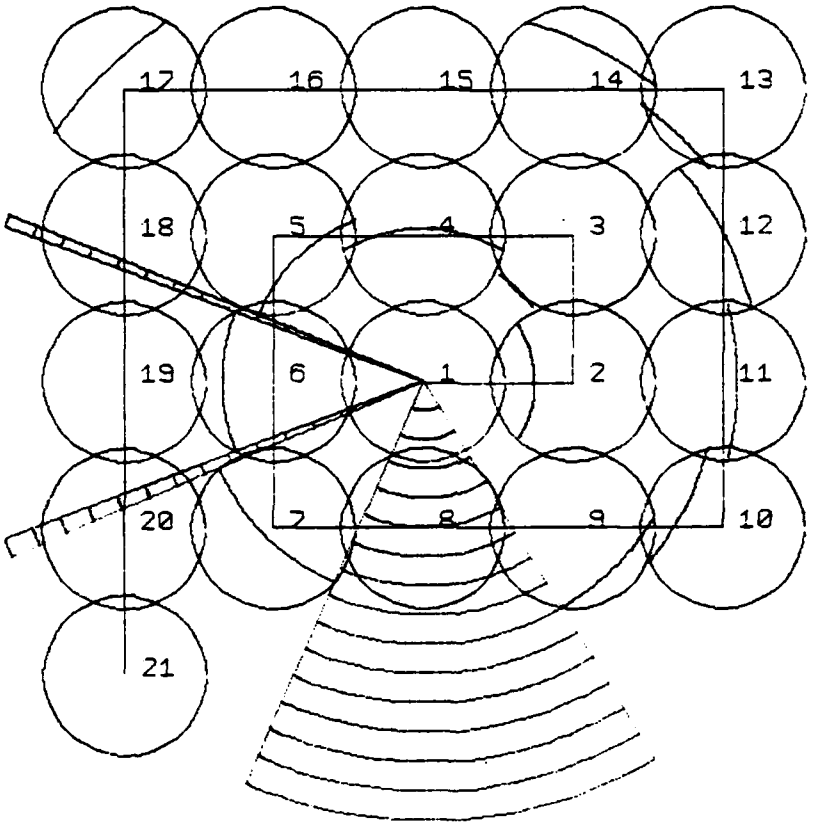
B(I)= CUMULATIVE ANGLE DETECTION COVERAGE AT X(I),Y(I).

B(2)= (0.00,0.57) (5.71,6.28)
B(3)= (0.00,0.57) (0.61,0.96) (5.71,6.28)
B(4)= (0.00,0.57) (0.61,0.96) (1.03,2.11) (5.71,6.28)
B(5)= (0.00,0.57) (0.61,0.96) (1.03,2.75) (5.71,6.28)
B(6)= (0.00,0.57) (0.61,0.96) (1.03,2.75) (2.77,3.51) (5.71,6.28)
B(7)= (0.00,0.57) (0.61,0.96) (1.03,2.75) (2.77,3.51) (3.55,4.30)
(5.71,6.28)
B(9)= (0.00,0.57) (0.61,0.96) (1.03,2.75) (2.77,3.51) (3.55,4.30)
(5.25,6.28)
B(12)= (0.00,0.96) (1.03,2.75) (2.77,3.51) (3.55,4.30) (5.25,6.28)
B(14)= (0.00,2.75) (2.77,3.51) (3.55,4.30) (5.25,6.28)

Execution Time = 0.46 Minutes

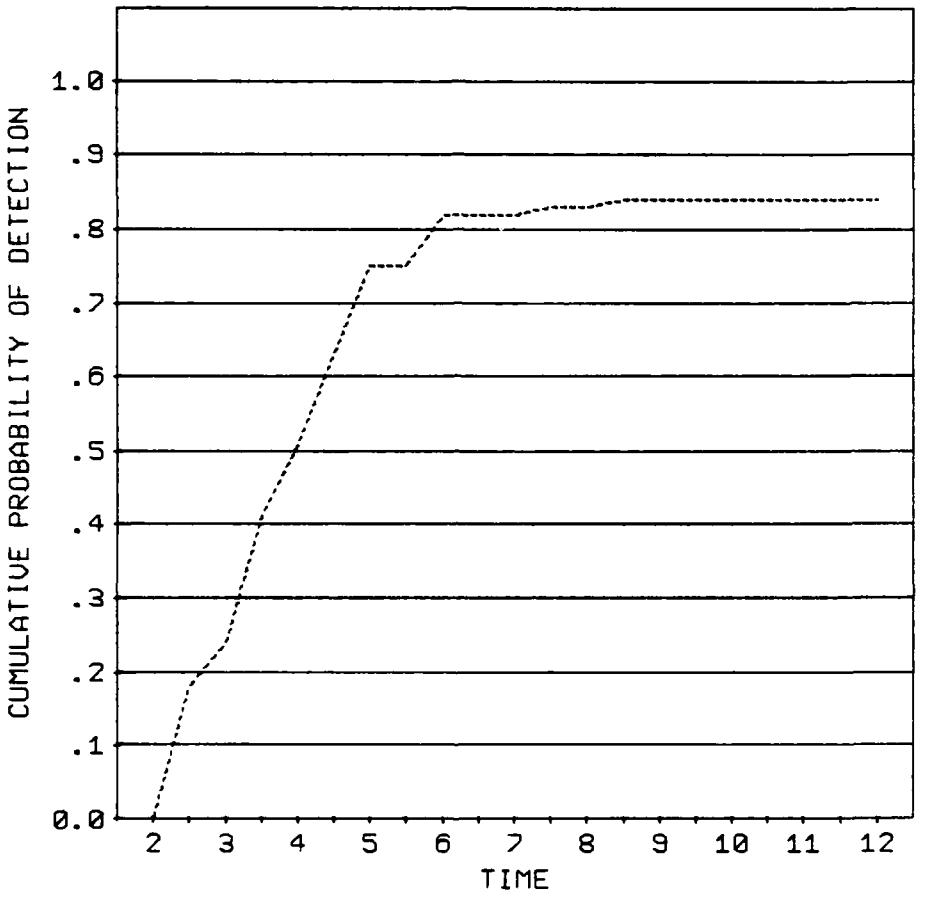
Example 1: BASIC Program Output - Expanding Spiral Search

Table 1



Example 1: Expanding Spiral Search Area Coverage

Figure 1.1



Example 1: Cumulative Probability of Detection vs Time

Figure 1.2

```
Andrus Stop-Look Detection Algorithm      05-24-1985


NUMBER OF (X,Y) =   21        I    : SEARCH POINT NUMBER
TIME LATE       =    2.00     X,Y  : SEARCH POINT COORDINATES
SEARCH VELOCITY =  290.00     TIME : TIME SEARCHER AT (X,Y)
TARGET VELOCITY =   30.00     TLCR : TARGET LOCATION CIRCLE RADIUS
DETECTION RANGE =   55.00     A1,A2: TARGET/SEARCH CIRCLE INTERSECTION ANGLES
                              C    : CUMULATIVE PROBABILITY OF DETECTION
```

I	X	Y	TIME	TLCR	A1	A2	C
1	0.00	0.00	2.00	60.00	0.00	0.00	0.00
2	100.00	0.00	2.34	70.34	5.72	0.56	0.18
3	100.00	100.00	2.69	80.69	0.00	0.00	0.18
4	0.00	100.00	3.03	91.03	0.99	2.15	0.36
5	-100.00	100.00	3.38	101.38	2.04	2.67	0.45
6	-100.00	0.00	3.72	111.72	2.63	3.66	0.60
7	-100.00	-100.00	4.07	122.07	3.53	4.32	0.71
8	0.00	-100.00	4.41	132.41	4.32	5.10	0.83
9	100.00	-100.00	4.76	142.76	5.11	5.89	0.93
10	200.00	-100.00	5.10	153.10	0.00	0.00	0.93
11	200.00	0.00	5.45	163.45	6.06	0.23	0.93
12	200.00	100.00	5.79	173.79	0.35	0.58	0.93
13	200.00	200.00	6.14	184.14	0.00	0.00	0.93
14	100.00	200.00	6.48	194.48	0.88	1.33	0.95
15	0.00	200.00	6.83	204.83	1.30	1.84	0.95
16	-100.00	200.00	7.17	215.17	1.79	2.28	0.95
17	-200.00	200.00	7.52	225.52	0.00	0.00	0.95
18	-200.00	100.00	7.86	235.86	2.44	2.91	0.95
19	-200.00	0.00	8.21	246.21	3.01	3.28	0.95
20	-200.00	-100.00	8.55	256.55	3.42	3.79	0.95
21	-200.00	-200.00	8.90	266.90	3.74	4.12	0.95

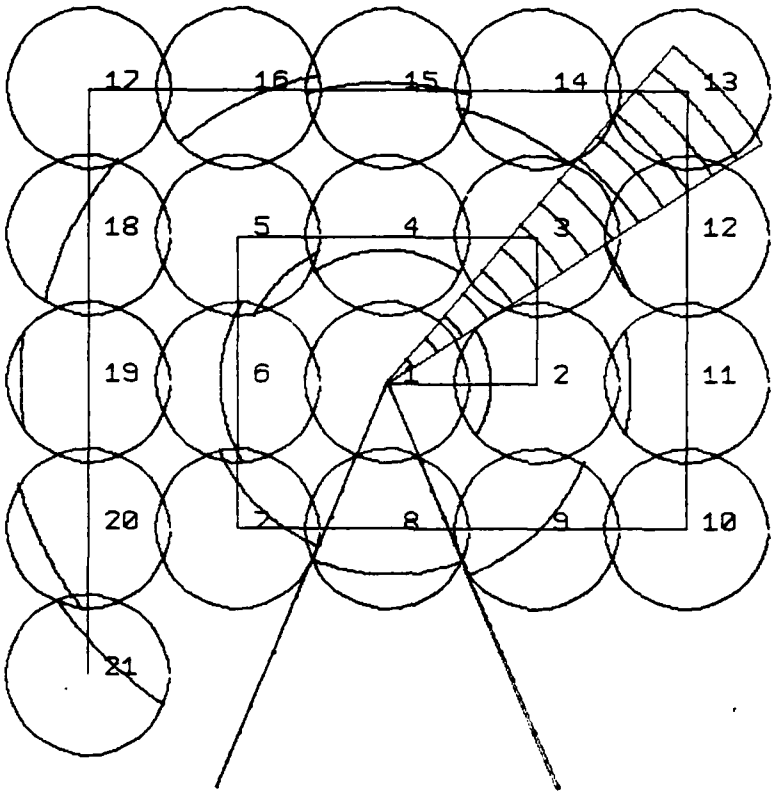
```
B( I )=  CUMULATIVE ANGLE DETECTION COVERAGE AT X(I),Y(I).

B( 2 )= (0.00,0.56)  (5.72,6.28)
B( 4 )= (0.00,0.56)  (0.99,2.15)  (5.72,6.28)
B( 5 )= (0.00,0.56)  (0.99,2.67)  (5.72,6.28)
B( 6 )= (0.00,0.56)  (0.99,3.66)  (5.72,6.28)
B( 7 )= (0.00,0.56)  (0.99,4.32)  (5.72,6.28)
B( 8 )= (0.00,0.56)  (0.99,4.32)  (4.32,5.10)  (5.72,6.28)
B( 9 )= (0.00,0.56)  (0.99,4.32)  (4.32,5.10)  (5.11,6.28)
B( 12 )= (0.00,0.58)  (0.99,4.32)  (4.32,5.10)  (5.11,6.28)
B( 14 )= (0.00,0.58)  (0.88,4.32)  (4.32,5.10)  (5.11,6.28)

Execution Time =    0.47 Minutes
```

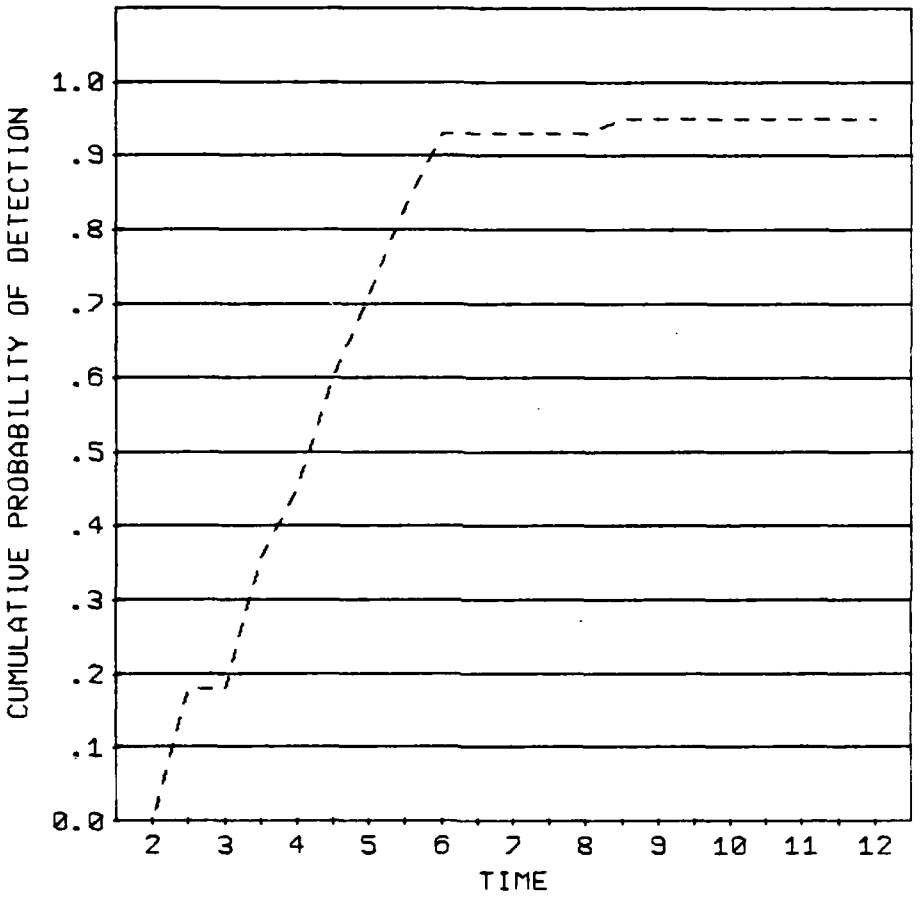
Example 2: BASIC Program Output - Expanding Spiral Search

Table 2



Example 2: Expanding Spiral Search Area Coverage

Figure 2.1



Example 2: Cumulative Probability of Detection vs Time

Figure 2.2

Andrus Stop-Look Detection Algorithm 05-24-1985

NUMBER OF (X,Y) = 20
TIME LATE = 2.00
SEARCH VELOCITY = 300.00
TARGET VELOCITY = 30.00
DETECTION RANGE = 55.00

I : SEARCH POINT NUMBER
X,Y : SEARCH POINT COORDINATES
TIME : TIME SEARCHER AT (X,Y)
TLCR : TARGET LOCATION CIRCLE RADIUS
A1,A2: TARGET/SEARCH CIRCLE INTERSECTION ANGLES
C : CUMULATIVE PROBABILITY OF DETECTION

I	X	Y	TIME	TLCR	A1	A2	C
1	-100.00	200.00	2.00	60.00	0.00	0.00	0.00
2	-100.00	100.00	2.33	70.00	0.00	0.00	0.00
3	-100.00	0.00	2.67	80.00	2.56	3.72	0.18
4	-100.00	-100.00	3.00	90.00	3.75	4.10	0.24
5	-100.00	-200.00	3.33	100.00	0.00	0.00	0.24
6	0.00	-200.00	3.67	110.00	0.00	0.00	0.24
7	0.00	-100.00	4.00	120.00	4.24	5.18	0.39
8	0.00	0.00	4.33	130.00	0.00	0.00	0.39
9	0.00	100.00	4.67	140.00	1.25	1.89	0.49
10	0.00	200.00	5.00	150.00	1.44	1.70	0.49
11	100.00	200.00	5.33	160.00	0.00	0.00	0.49
12	100.00	100.00	5.67	170.00	0.48	1.09	0.59
13	100.00	0.00	6.00	180.00	0.00	0.00	0.59
14	100.00	-100.00	6.33	190.00	5.34	5.66	0.64
15	100.00	-200.00	6.67	200.00	4.94	5.41	0.66
16	200.00	-200.00	7.00	210.00	0.00	0.00	0.66
17	200.00	-100.00	7.33	220.00	5.57	6.07	0.73
18	200.00	0.00	7.67	230.00	6.07	0.22	0.80
19	200.00	100.00	8.00	240.00	0.24	0.69	0.84
20	200.00	200.00	8.33	250.00	0.62	0.95	0.84

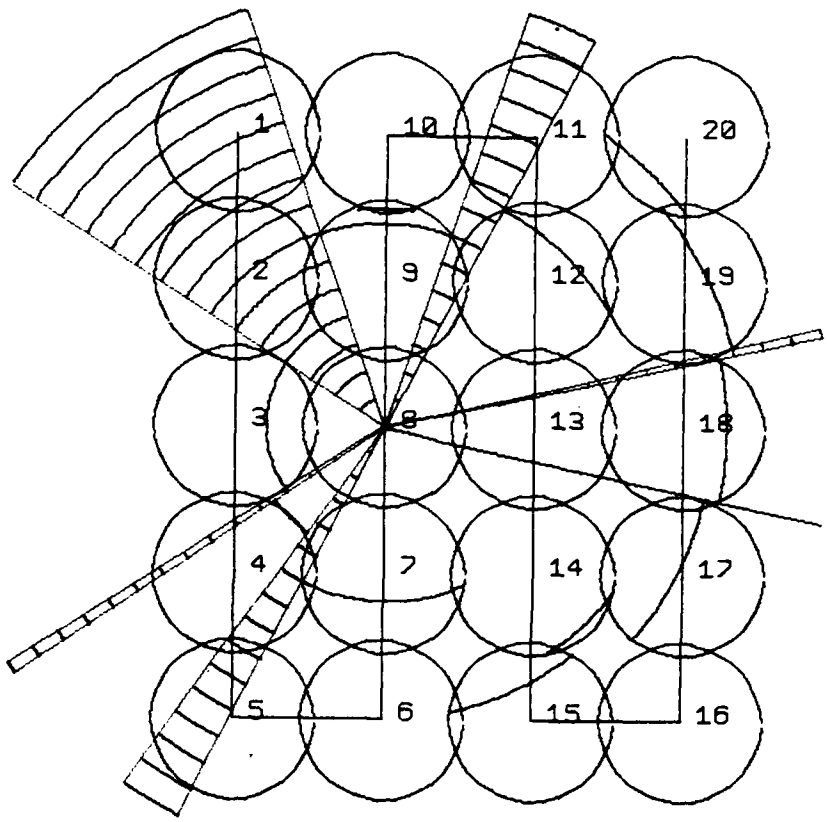
B(I) = CUMULATIVE ANGLE DETECTION COVERAGE AT X(I),Y(I).

B(3) = (2.56,3.72)
B(4) = (2.56,3.72) (3.75,4.10)
B(7) = (2.56,3.72) (3.75,4.10) (4.24,5.18)
B(9) = (1.25,1.89) (2.56,3.72) (3.75,4.10) (4.24,5.18)
B(12) = (0.48,1.09) (1.25,1.89) (2.56,3.72) (3.75,4.10) (4.24,5.18)
B(14) = (0.48,1.09) (1.25,1.89) (2.56,3.72) (3.75,4.10) (4.24,5.18)
(5.34,5.66)
B(15) = (0.48,1.09) (1.25,1.89) (2.56,3.72) (3.75,4.10) (4.24,5.66)
B(17) = (0.48,1.09) (1.25,1.89) (2.56,3.72) (3.75,4.10) (4.24,6.07)
B(18) = (0.00,0.22) (0.48,1.09) (1.25,1.89) (2.56,3.72) (3.75,4.10)
(4.24,6.07) (6.07,6.28)
B(19) = (0.00,0.22) (0.24,1.09) (1.25,1.89) (2.56,3.72) (3.75,4.10)
(4.24,6.07) (6.07,6.28)

Execution Time = 0.49 Minutes

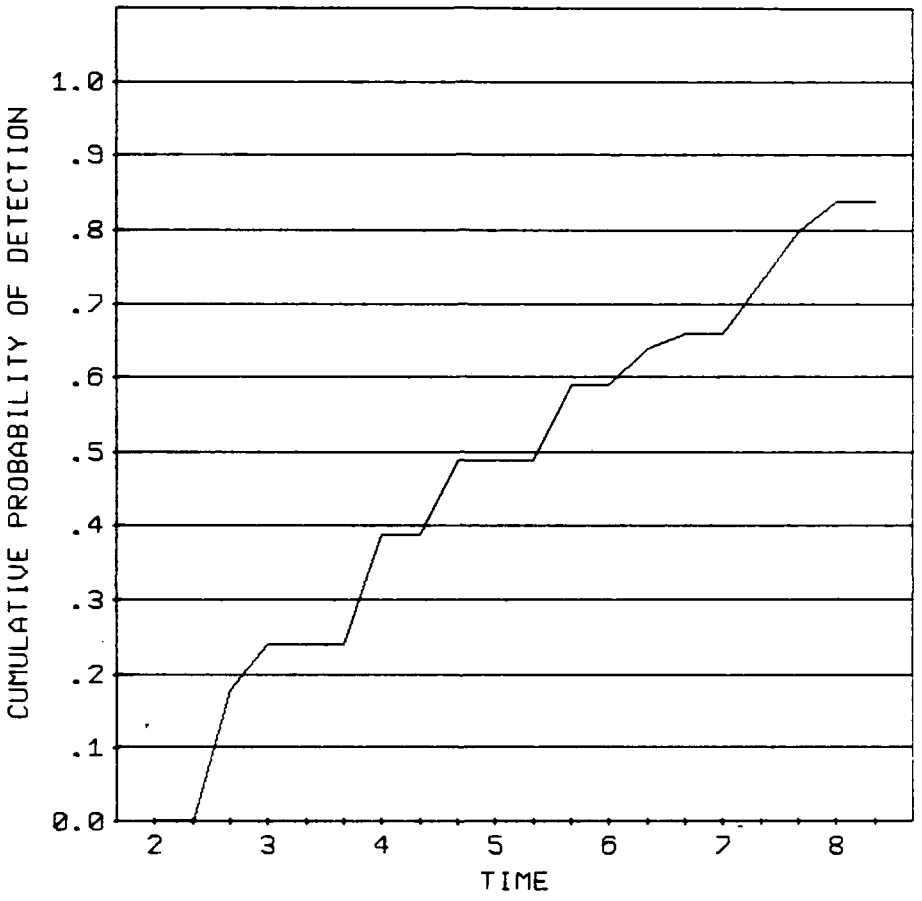
Example 3: BASIC Program Output - Grid Search

Table 3



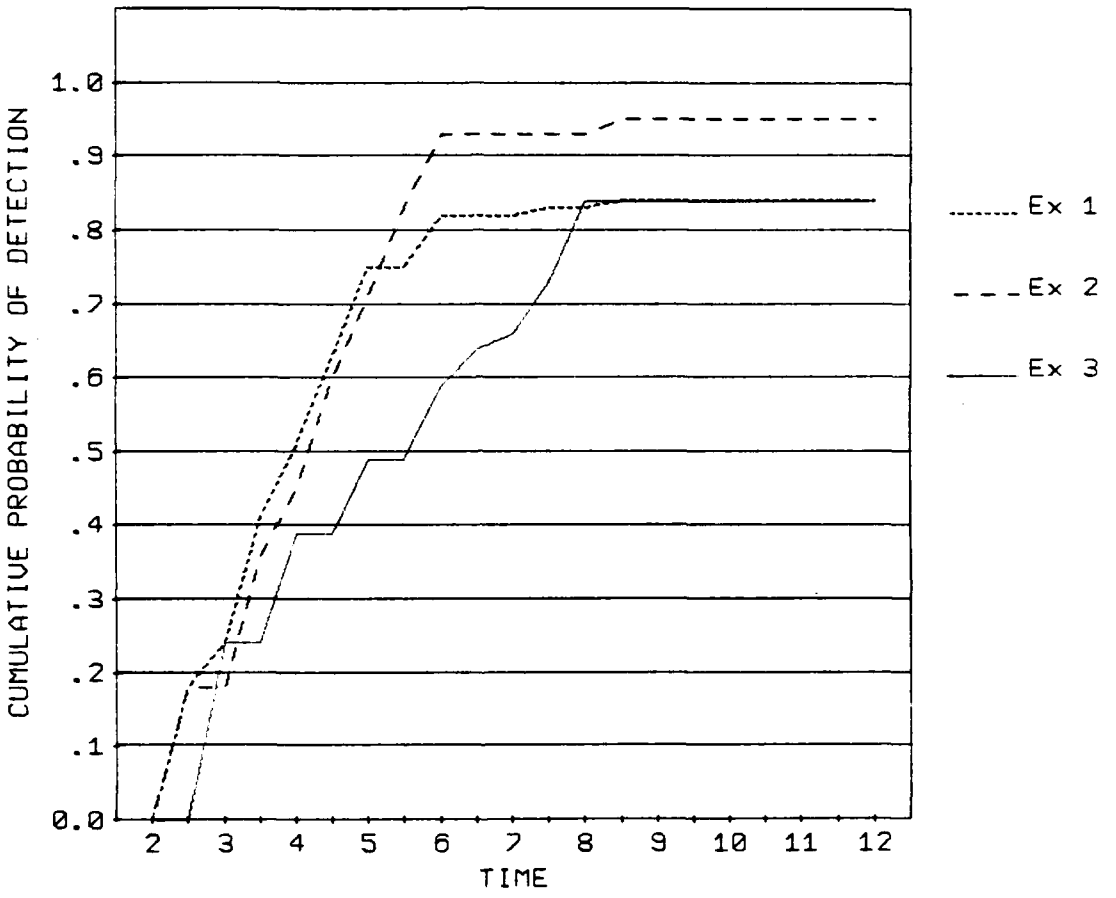
Example 3: Grid Search Area Coverage

Figure 3.1



Example 3: Cumulative Probability of Detection vs Time

Figure 3.2



Cumulative Probability of Detection vs Time for Examples 1,2,3

Figure 4

10. Assumption Modifications:

The assumptions for this version of the Stop and Look algorithm can be changed with minor modifications to the algorithm procedures and the program contained in Appendix A. Examples of these assumption changes are:

a. Searcher velocity: Searcher velocity does not have to be constant. What is required is that the time at each search point be known.

b. Target course: Target course does not have to range over (0,2*Pi) or be uniformly distributed. If target course is distributed uniformly then C(i) is computed using the assigned target range. If the target course values are not uniformly distributed then C(i) would be computed by weighting the components of B(i) according to the target course distribution.

c. Target velocity: Target velocity does not have to be constant. What is required is that target velocity as a function of time is known.

Other assumption changes can be suggested. For example it is possible to assume distributions for target and searcher velocities and compute C(i) by weighting the B(i) components according to the velocity distributions. However, as the assumption changes become more complex the algorithm modifications increase in complexity.

Appendix A: BASIC Program Listing:

```
10 'ANDRUS STOP AND LOOK ALGORITHM: VERSION 1.0: 26 OCTOBER 1984
20 DEFDBL A-H,P-Z: DEFINT I-O: KEY OFF
30 CLS: DEF FNACOS(A)=1.570796-ATN(A/SQR(1-A*A)): PI=3.1415926535897938
40 DIM S(51),ST(51),TR(51),TI(51),A1(102),A2(102),X(51),Y(51),B1(102),
     B2(102),C(51)
50 A1$="Stop and Look Detection Algorithm":A2$="Written by:"
60 A3$="Professor Alvin Andrus, Code 55As":A4$="Naval Postgraduate School"
70 A5$="Monterey, Ca. 93940":A6$="To Continue use ENTER key"
80 LOCATE 10,40-LEN(A1$)/2: PRINT A1$: LOCATE 12,40-LEN(A2$)/2: PRINT A2$
90 LOCATE 14,40-LEN(A3$)/2: PRINT A3$: LOCATE 16,40-LEN(A4$)/2: PRINT A4$
100 LOCATE 18,40-LEN(A5$)/2: PRINT A5$: LOCATE 24,40-LEN(A6$)/2: PRINT A6$;
110 INPUT "",A$: CLS: LOCATE 3,1
120 PRINT"          This program computes the probability of detection for a"
130 PRINT"          searcher looking discretely for an evading target.":PRINT
140 PRINT"     Assumptions are:": PRINT
150 PRINT"     1) The datum is located at (0,0)."
160 PRINT"     2) Detection is deterministic, i.e., cookie cutter."
170 PRINT"     3) The searcher begins searching at time late with constant velocity."
180 PRINT"     4) The target leaves datum on a random course with constant velocity."
190 PRINT"     5) The searcher stops and looks for the target at specified points."
200 PRINT
210 PRINT"     Input Requirements are:":PRINT
220 PRINT"     1) The time late."
230 PRINT"     2) The target speed."
240 PRINT"     3) The searcher speed."
250 PRINT"     4) The detection range."
260 PRINT"     5) The stop and look Points: X(I),Y(I). I<=50.": PRINT
270 PRINT"     The units used for speed, time and range must be consistent."
280 LOCATE 24,40-LEN(A6$)/2: PRINT A6$;: INPUT "",A$: CLS: LOCATE 5,1
290 PRINT"          To Create a New Input File:              Type N or n."
300 PRINT"          To Use an Existing Input File:           Type E or e."
310 PRINT"          To Change and Use an Existing Input File: Type C or c."
320 A$=INKEY$
330 IF A$="C" OR A$="c" OR A$="E" OR A$="e" OR A$="N" OR A$="n" THEN 350
340 GOTO 320
350 PRINT: INPUT "          Input File Name  = "; INFILE$
360 PRINT:INPUT "          Output File Name = "; OUT1$: OUT2$=OUT1$+".1": PRINT
370 IF A$="E" OR A$="e" OR A$="C" OR A$="c" THEN 680
380 PRINT"     If value is correct use ENTER key."
390 PRINT"     If value is incorrect enter new value.":PRINT
400 PRINT"          Searcher speed  =";SV;:INPUT SV$:IF SV$<>"" THEN SV=VAL(SV$)
410 PRINT"          Target speed    =";TV;:INPUT TV$: IF TV$<>"" THEN TV=VAL(TV$
420 PRINT"          Time late       =";TL;:INPUT TL$: IF TL$<>"" THEN TL=VAL(TL$
430 PRINT"          Detection range =";R;:INPUT R$: IF R$<>"" THEN R=VAL(R$)
440 PRINT"          Number of stop and look points (<=50) =";N;
450 INPUT N$: IF N$<>"" THEN N=VAL(N$)
460 CLS: IF N>=10 THEN 530
470 FOR I=1 TO N
480     LOCATE I+2, 1: PRINT "X";RIGHT$(STR$(I),1);"=";X(I);:INPUT X$
490     IF X$<>"" THEN X(I)=VAL(X$)
500     LOCATE I+2,40: PRINT "Y";RIGHT$(STR$(I),1);"=";Y(I);:INPUT Y$
510     IF Y$<>"" THEN Y(I)=VAL(Y$)
```

```
520 NEXT I: GOTO 650
530 FOR I=1 TO 9
540     LOCATE I+2, 1: PRINT "X";RIGHT$(STR$(I),1);"=";X(I);:INPUT X$
550     IF X$<>"" THEN X(I)=VAL(X$)
560     LOCATE I+2,40: PRINT "Y";RIGHT$(STR$(I),1);"=";Y(I);:INPUT Y$
570     IF Y$<>"" THEN Y(I)=VAL(Y$)
580 NEXT I
590 FOR I=10 TO N
600     LOCATE I+2, 1: PRINT "X";RIGHT$(STR$(I),2);"=";X(I);:INPUT X$
610     IF X$<>"" THEN X(I)=VAL(X$)
620     LOCATE I+2,40: PRINT "Y";RIGHT$(STR$(I),2);"=";Y(I);:INPUT Y$
630     IF Y$<>"" THEN Y(I)=VAL(Y$)
640 NEXT I
650 OPEN INFILE$ FOR OUTPUT AS 1
660 PRINT#1,N;TL;SV;TV;R: FOR I=1 TO N: PRINT#1,X(I);Y(I): NEXT I
670 CLOSE# 1: GOSUB 1500
680 OPEN INFILE$ FOR INPUT AS 1
690 INPUT#1,N,TL,SV,TV,R: FOR I=1 TO N: INPUT#1,X(I),Y(I): NEXT I
700 CLOSE# 1: IF A$="C" OR A$="c" THEN 380 ELSE GOSUB 1500
710 BT=TIMER: S(1)=SQR(X(1)^2+Y(1)^2): ST(1)=0: TI(1)=TL: TR(1)=TL*TV
720 FOR I=2 TO N
730      S(I)= SQR(X(I)^2+Y(I)^2)
740      ST(I)=ST(I-1)+SQR((X(I)-X(I-1))^2+(Y(I)-Y(I-1))^2)
750      TI(I)= ST(I)/SV+TL: TR(I)=TI(I)*TV
760 NEXT I
770 FOR I=1 TO N
780     IF R<=ABS(TR(I)-S(I)) THEN A1(I)=0: A2(I)=0: GOTO 910
790     IF R<S(I)+TR(I) THEN 810
800     A1(I)=0: A2(I)=2*PI: GOTO 910
810     IF X(I)=0 AND Y(I)>0 THEN A=PI/2: GOTO 860
820     IF X(I)=0 AND Y(I)<0 THEN A=3*PI/2: GOTO 860
830     A=ATN(Y(I)/X(I))
840     IF X(I)<0 THEN A=A+PI
850     IF X(I)>0 AND Y(I)<0 THEN A=A+2*PI
860     B=(R^2-S(I)^2-TR(I)^2)/(-2*S(I)*TR(I)) :A1(I)=FNACOS(B)
870     A2(I)=2*PI-A1(I)+A: A1(I)=A1(I)+A
880     IF A1(I)>2*PI THEN A1(I)=A1(I)-2*PI: GOTO 880
890     IF A2(I)>2*PI THEN A2(I)=A2(I)-2*PI: GOTO 890
900     IF A<A1(I) OR A>A2(I) THEN SWAP A1(I),A2(I)
910 NEXT I
920 K=0
930 FOR I=1 TO N
940    IF A1(I)=A2(I) THEN 1470
950    IF K <> 0 THEN 980
960    IF A1(I)< A2(I) THEN B1(1)=A1(I):B2(1)=A2(I):C(I)=A2(I)-A1(I):K=1:
          GOTO 1470
970    B1(1)=0:B2(1)=A2(I):B1(2)=A1(I):B2(2)=2*PI:C(I)=A2(I)-A1(I)+2*PI:K=2:
          GOTO 1470
980    IF A1(I)<A2(I) THEN A=A1(I): B=A2(I): FLAG=1: GOTO 1000
990    A=0: B=A2(I): C=A1(I): D=2*PI: FLAG=2: GOTO 1000
1000   FOR J=1 TO K
1010      IF A<=B1(J) THEN IX=1: IX1=J: GOTO 1050
1020      IF A<=B2(J) THEN IX=2: IX1=J: GOTO 1050
1030   NEXT J
1040   B1(K+1)=A: B2(K+1)=B: C(I)=C(I)+B-A: K=K+1: GOTO 1460
```

```
1050    FOR J=1 TO K
1060       IF B<= B1(J) THEN IY=1: IY1=J: GOTO 1100
1070       IF B<= B2(J) THEN IY=2: IY1=J: GOTO 1100
1080    NEXT J
1090    IY=3
1100    IF IX=1 AND IY=1 AND IX1=IY1 THEN 1200
1110    IF IX=1 AND IY=1 AND IX1<>IY1 THEN 1230
1120    IF IX=1 AND IY=2 AND IX1=IY1 THEN 1260
1130    IF IX=1 AND IY=2 AND IX1<>IY1 THEN 1280
1140    IF IX=2 AND IY=1 AND IX1=IY1 THEN 1320
1150    IF IX=2 AND IY=1 AND IX1<>IY1 THEN 1330
1160    IF IX=2 AND IY=2 AND IX1=IY1 THEN 1460
1170    IF IX=2 AND IY=2 AND IX1<>IY1 THEN 1370
1180    IF IX=1 AND IY=3 THEN 1420
1190    IF IX=2 AND IY=3 THEN 1440
1200    C(I)=C(I)+B-A
1210    FOR L=K TO IX1 STEP -1: B1(L+1)=B1(L): B2(L+1)=B2(L): NEXT L
1220    B1(IX1)=A: B2(IX1)=B: K=K+1: GOTO 1460
1230    C(I)=C(I)+B-A: FOR L=IX1 TO IY1-1: C(I)=C(I)+B1(L)-B2(L): NEXT L
1240    FOR L=1 TO K-IY1: B1(IX1+L)=B1(IY1+L): B2(IX1+L)=B2(IY1+L): NEXT L
1250    B1(IX1)=A: B2(IX1)=B: K=K-IY1+IX1: GOTO 1460
1260    C(I)=C(I)+B1(IX1)-A
1270    B1(IX1)=A: GOTO 1460
1280    C(I)=C(I)+B1(IY1)-A: FOR L=IX1 TO IY1-1: C(I)=C(I)+B1(L)-B2(L): NEXT L
1290    B1(IX1)=A: B2(IX1)=B2(IY1)
1300    FOR L=1 TO K=IY1: B1(IX1+L)=B1(IY1+L): B2(IX1+L)=B2(IY1+L): NEXT L
1310    K=K-IY1+IX1: GOTO 1460
1320    CLS: PRINT "ERROR IN ALGORITHM LOGIC: PROGRAM EXIT": END
1330    C(I)=C(I)+B-B2(IX1)
1340    FOR L=IX1+1 TO IY1-1: C(I)=C(I)+B1(L)-B2(L): NEXT L
1350    FOR L=1 TO K+1-IY1: B1(IX1+L)=B1(IY1+L-1): B2(IX1+L)=B2(IY1+L-1): NEXT L
1360    B2(IX1)=B: K=K-IY1+IX1+1: GOTO 1460
1370    C(I)=C(I)+B1(IY1)-B2(IX1)
1380    FOR L=IX1+1 TO IY1-1: C(I)=C(I)+B1(L)-B2(L): NEXT L
1390    B2(IX1)=B2(IY1)
1400    FOR L=1 TO K-IY1: B1(IX1+L)=B1(IY1+L): B2(IX1+L)=B2(IY1+L): NEXT L
1410    K=K-IY1+IX1: GOTO 1460
1420    C(I)=C(I)+B-A: FOR L=IX1 TO K: C(I)=C(I)+B1(L)-B2(L): NEXT L
1430    B1(IX1)=A: B2(IX1)=B: K=IX1: GOTO 1460
1440    C(I)=C(I)+B-B2(IX1): FOR L=IX1+1 TO K: C(I)=C(I)+B1(L)-B2(L): NEXT L
1450    B2(IX1)=B: K=IX1: GOTO 1460
1460    IF FLAG=1 THEN 1470 ELSE A=C: B=D: FLAG=1: GOTO 1000
1470    IF I=1 THEN C(1)=C(1)/(2*PI) ELSE C(I)=C(I)/(2*PI)+C(I-1)
1480    GOSUB 1670
1490 NEXT I: CLOSE#2: GOTO 1770
1500 OPEN OUT1$ FOR OUTPUT AS 1: OPEN OUT2$ FOR OUTPUT AS 2
1510 PRINT#1, "Andrus Stop-Look Detection Algorithm",DATE$: PRINT#1,:PRINT#1,
1520 PRINT#1,"NUMBER OF (X,Y) =";:PRINT#1, USING"#####";N;
1530 PRINT#1,,"  I     : SEARCH POINT NUMBER"
1540 PRINT#1,"TIME LATE       =";:PRINT#1, USING"#####.##";TL;
1550 PRINT#1,,"  X,Y   : SEARCH POINT COORDINATES"
1560 PRINT#1,"SEARCH VELOCITY =";:PRINT#1, USING"#####.##";SV;
1570 PRINT#1,,"  TIME  : TIME SEARCHER AT (X,Y)"
1580 PRINT#1,"TARGET VELOCITY =";:PRINT#1, USING"#####.##";TV;
1590 PRINT#1,,"  TLCR  : TARGET LOCATION CIRCLE RADIUS"
```

```
1600 PRINT#1,"DETECTION RANGE =";:PRINT#1, USING"#####.##";R;
1610 PRINT#1,,"  A1,A2: TARGET/SEARCH CIRCLE INTERSECTION ANGLES"
1620 PRINT#1,,,"  C    : CUMULATIVE PROBABILITY OF DETECTION"
1630 PRINT#1,: PRINT#1,: PRINT#2,
1640 PRINT#1,"  I       X         Y        TIME     TLCR      A1        A2      C"
1650 PRINT#2, " B( I )=  CUMULATIVE ANGLE DETECTION COVERAGE AT X(I),Y(I)."
1660 PRINT#1,: RETURN 710
1670 PRINT#1, USING "###";I;
1680 PRINT#1, USING "######.##";X(I),Y(I),TI(I),TR(I),A1(I),A2(I),C(I)
1690 IF A1(I)<>A2(I) AND C(I)<>C(I-1) THEN 1700 ELSE RETURN
1700 L=1: PRINT#2,: PRINT#2," B(";I;")= ";: IF I<10 THEN PRINT#2, " ";
1710 FOR J=1 TO K
1720   PRINT#2, "(";: PRINT#2, USING "#.##";B1(J);: PRINT#2, ",";:
       PRINT#2, USING "#.##";B2(J);: PRINT#2, ")  ";
1730   IF L=5 AND J<>K THEN PRINT#2,
1740   L=L+1: IF L>5 AND J<>K THEN PRINT#2, "          ";: L=1
1750 NEXT J
1760 RETURN
1770 OPEN OUT2$ FOR INPUT AS 2
1780 WHILE NOT EOF(2)
1790    LINE INPUT#2, A$
1800    PRINT#1, A$
1810 WEND
1820 ET=TIMER: IF ET<BT THEN A=24*60*60-BT+ET ELSE A=ET-BT
1830 PRINT#1,:PRINT#1,"Execution Time = ";: PRINT#1, USING "####.##";A/60;:
     PRINT#1, " Minutes"
1840 CLOSE#1: OPEN OUT1$ FOR INPUT AS 1
1850 CLS
1860 FOR I=1 TO 23
1870    IF EOF(1) THEN 1920
1880    LINE INPUT#1, A$
1890    PRINT A$
1900 NEXT I: LOCATE 25,24: PRINT "PRESS ENTER KEY TO SCROLL SCREEN";
1910 A$=INKEY$: IF A$="" THEN 1910 ELSE CLS: GOTO 1860
1920 CLOSE#1: CLOSE#2: KILL OUT2$: END
```

Appendix B: Computation Procedures:

The computations used in the program of Appendix A for the Stop and Look algorithm are:

a. S(i) = SQR(X(i)^2+Y(i)^2).

S(i) is the distance from (0,0) to (X(i),Y(i)).

b. ST(i) = SUM [SQR(X(j)-X(j-1))^2+(Y(j)-Y(j-1))^2]

for j =i to i and ST(1)=0.

ST(i) is the total distance moved by the searcher from (X(1),Y(1)) to (X(i),Y(i)).

c. TI(i) = ST(i)/SV+TL.

T(i) is the time the searcher is at (X(i),Y(i)). SV is the search velocity and TL is the time late.

d. TR(i) = TI(i)*TV.

TR(i) is the radius of the target location circle. TV is the target velocity.

e. r = TI(i)*TV.

The equation for the target location circle at time TI(i). r is the radial coordinate.

f. R^2 = r^2-2*r*S(i)*COS(A-ATAN(Y(i)/X(i)))+S(i)^2.

The equation for the searcher detection circle at time TI(i). A is angle coordinate. R is the detection range.

e. A(i)=(A1(i),A2(i))

=ACOS(R^2-S(i)^2-TR(i)^2)/(-2*S(i)*TR(i))+ATAN(Y(i)/X(i)).

A(i) is the pair of polar angle coordinates for the intersection of the target location circle and the searcher detection circle at time TI(i).

DISTRIBUTION LIST

	NO. OF COPIES
Defense Technical Information Center Cameron Station Alenandria, VA 22314	2
Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	2
Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93940	1
Office of Naval Research 800 North Quincy Street Arlington, VA 22217	1
Chief of Naval Operations Code OP-953 Washington, DC 20350	1
Navy Tactical Support Activity P.O. Box 1042 Silver Spring, MD 20910	1
Naval Air Development Center ATTN: Dr. M. Leonardo Warminster, PA 18974	1
COMPATWINGSLANT Code N7 Naval Air Station Brunswick, ME 04011	1
COMPATWINGSLANT Code 51 and 532 Naval Air Station Moffett Field, CA 94035	1
U. S. Naval Academy Department of Operations Research Annapolis, MD 21402	1
Commander Surface Warfare Development Group Naval Amphibious Base, Little Creek Norfolk, VA 23521	1

COMTHIRDFLEET 1
Code N7
Pearl Harbor
Honolulu, HI 96818

COMSECONDFLEET 1
Code N7
Norfolk, VA 23511

Naval Ocean System Center 1
Code 16
San Diego, CA 92132

COMTRAGRUPAC 1
ATTN: Mr. Terry Foust
200 Catalina Blvd.
San Diego, CA 92147

COMSUBPAC 1
ATTN: TAC D & E
Pearl Harbor
Honolulu, HI 96818

COMASWWINGSPAC 1
ATTN: TAC D & E
North Island Naval Air Station
San Diego, CA 92135

Professor A. Andrus 50
Code 55As
Naval Postgraduate School
Monterey, Ca 93940

END

FILMED

10-85

DTIC